Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 30, no. 1, pp. 47–63 (2026)
DOI: 10.7155/jgaa.v30i1.2967

Largest Common Subgraph of Two Forests

Dieter Rautenbach *Florian Werner*

Institute of Optimization and Operations Research,
Ulm University, Ulm, Germany

Submitted: Sept. 2024 Accepted: December 2025 Published: March 2026

Article type: Regular paper Communicated by: H. Meyerhenke

Abstract. A common subgraph of two graphs G_1 and G_2 is a graph that is isomorphic to subgraphs of G_1 and G_2. In the largest common subgraph problem the task is to determine a common subgraph for two given graphs G_1 and G_2 that has maximum possible number of edges $\text{lcs}(G_1, G_2)$. This natural problem generalizes the well-studied graph isomorphism problem, has many applications, and remains NP-hard even restricted to unions of paths. We present a simple 4-approximation algorithm for forests, and, for every fixed $\epsilon \in (0, 1)$, we show that, for two given forests F_1 and F_2 of order at most n, one can determine in polynomial time a common subgraph F of F_1 and F_2 with at least $\text{lcs}(F_1, F_2) - \epsilon n$ edges. Restricted to instances with $\text{lcs}(F_1, F_2) \geq cn$ for some fixed positive c, this yields a polynomial time approximation scheme. Our approach relies on the approximation of the given forests by structurally simpler forests that are composed of copies of only $O(\log(n))$ different starlike rooted trees and iterative quantizations of the options for the solutions.

1 Introduction

We consider finite, simple, and undirected graphs, and use standard terminology. A graph H is a *subgraph* of a graph G if H arises from G by removing vertices and edges and a subgraph of G is *spanning* if it contains all vertices of G. If H is a subgraph of a graph G, we write $H \subseteq G$. The order $n(G)$ and the size $m(G)$ of a graph G are the numbers of its vertices and edges, respectively. For a positive integer k, let $[k]$ be the set of positive integers at most k and let $[k]_0 = \{0\} \cup [k]$.

In their seminal list of NP-complete problems, Garey and Johnson [7] mention the following decision problem as [GT49].

Largest Common Subgraph
Instance: Two graphs G and H, and a positive integer K.
Question: Are there spanning subgraphs G' of G and H' of H that have at least K edges and are isomorphic?

E-mail addresses: dieter.rautenbach@uni-ulm.de (Dieter Rautenbach) florian.werner@uni-ulm.de (Florian Werner)

Note that two graphs G and H are isomorphic if and only if G, H, and $K = m(G)$ form a yes-instance of Largest Common Subgraph, that is, this problem generalizes the graph isomorphism problem. It was proposed by Bokhari [4] within the context of array processing and has various applications ranging from molecular chemistry [14] to pattern matching [15]. Since the subgraphs G' and H' in the problem statement are obtained by removing edges only, Largest Common Subgraph makes sense only for graphs G and H of the same order. Its NP-completeness follows easily from the NP-completeness of Clique, which is [GT19] in [7].

In their comment to [GT49], Garey and Johnson claim that Largest Common Subgraph can be solved in polynomial time if both G and H are trees, for which they cite a private communication by Edmonds and Matula from 1975. Grohe, Rattan, and Woeginger [8] show that — unless $P = NP$ — this claim is false by a reduction from 3-partition, which is [SP15] in [7]. In fact, let I be an instance of 3-partition that consists of $3m$ positive integers $a_1, \dots, a_{3m}$ with $A/4 < a_i < A/2$ for each $i \in [3m]$, where $A = \frac{1}{m}(a_1 + \cdots + a_{3m})$. Recall that the question for I is whether there is a partition of $[3m]$ into m sets $I_1, \dots, I_m$ each containing exactly three elements such that $\sum_{j \in I_i} a_j = A$ for each $i \in [m]$. Now, it is easy to see that I is a yes-instance of 3-partition if and only if G, H, and K form a yes-instance of Largest Common Subgraph, where G is the disjoint union of $3m$ paths of orders $a_1, \dots, a_{3m}$, H is the disjoint union of m paths each of order A, and K is the size of G. Since 3-partition is NP-complete in the strong sense [7], it follows that Largest Common Subgraph remains NP-complete when restricted to instances where G and H are unions of paths. A simple modification yields the following.

Theorem (Grohe, Rattan, and Woeginger, Theorem 8 in [8]) Largest Common Subgraph *remains NP-complete when restricted to instances where G and H are both trees.*

Possibly, Edmonds and Matula had a different problem in mind, namely to determine the largest common subtree of two given trees. This is supported by Matula's discussion in [13]. In [1] Akutsu gives details for a simple efficient dynamic programming algorithm solving this problem using the maximum weight bipartite matching algorithm as a subroutine. Exploiting the structure of the specific matching instances that arise, Droschinsky et al. [5] present a faster algorithm. Many variations of Largest Common Subgraph have been considered in view of their relevance for certain applications. The variations involve the restriction to connected common subgraphs, vertex/edge labels that have to be respected, and topological notions of subgraphs, cf. [2,3,6,9,12,17] and references therein. Approximation algorithms and hardness of approximation were also studied [10,11].

In the present paper, we consider approximation algorithms for the maximization version of Largest Common Subgraph restricted to forests. Let a graph H be a *common subgraph* of two graphs G_1 and G_2 if H is isomorphic to a subgraph of G_1 as well as to a subgraph of G_2. Let $\text{lcs}(G_1, G_2)$ be the largest size $m(H)$ of a common subgraph H of G_1 and G_2. Note that we ignore the restriction to spanning subgraphs from the statement of Largest Common Subgraph as it is not essential. In fact, if H is a common subgraph of two graphs G_1 and G_2, which are both of the same order n, then adding $n - n(H)$ isolated vertices to H yields a graph H' of the same size as H that is isomorphic to spanning subgraphs of G_1 and G_2.

While Largest Common Subgraph remains NP-complete when restricted to unions of paths, we show with Lemma 1 below that it can be solved efficiently when restricted to unions of stars. This yields a simple 4-factor approximation algorithm.

Theorem 1. *For two given forests F_1 and F_2 of order n, one can determine a common subgraph F of F_1 and F_2 with*

$$m(F) \geq \frac{1}{4}\text{lcs}(F_1, F_2)$$

in time $n^{O(1)}$.

A natural next goal would be a polynomial time approximation scheme (PTAS) for Largest Common Subgraph restricted to forests. Our second result yields a PTAS when restricted to instances (F_1, F_2), where F_1 and F_2 are forests of order n and $\text{lcs}(F_1, F_2) \geq cn$ for some fixed positive c.

Theorem 2. *For every $\epsilon \in (0, 1)$, there is some $k \in \mathbb{N}$ with the following property: For two given forests F_1 and F_2 of order n, one can determine a common subgraph F of F_1 and F_2 with*

$$m(F) \geq \text{lcs}(F_1, F_2) - \epsilon n$$

in time $O(n^k)$.

Our approach for Theorem 2 is as follows. Firstly, removing a small fraction of all edges, we approximate the given forests F_1 and F_2 by simpler forests that are composed of copies of $O(\log n)$ different trees that are structurally close to stars. In particular, each component K of the approximating forests has a root r of controlled degree and the components of $K - r$ are of bounded order. Secondly, we show how to solve the largest common subgraph problem approximately on such simpler instances. For this approximate solution, we reduce the number of options that need to be considered at several stages during our algorithm by suitable quantization.

The following two sections contain proofs of our results and auxiliary statements.

2 Proof of Theorem 1

In this section, we show Theorem 1.

Our first lemma implies that, for two given unions F_1 and F_2 of stars, a common subgraph F of F_1 and F_2 of maximum size $\text{lcs}(F_1, F_2)$ can be found efficiently. The key observation is the following simple inequality: For every $a, a', b, b' \in \mathbb{N}_0$ with $a < a'$ and $b < b'$, we have

$$\min\{a, b'\} + \min\{a', b\} \leq \min\{a, b\} + \min\{a', b'\}. \tag{1}$$

The inequality (1) follows easily by considering all possible non-decreasing orderings of a, b, a', b'.

Lemma 1. *Let $a_1, \ldots, a_\ell$ and $b_1, \ldots, b_\ell$ be two non-decreasing sequences of non-negative integers. If F_1 is the disjoint union of ℓ stars of orders $a_1 + 1, \ldots, a_\ell + 1$, and F_2 is the disjoint union of stars of orders $b_1 + 1, \ldots, b_\ell + 1$, then*

$$\text{lcs}(F_1, F_2) = \sum_{i=1}^{\ell} \min\{a_i, b_i\}.$$

Proof: Let F be a common subgraph of F_1 and F_2 with $m(F) = \text{lcs}(F_1, F_2)$. By renaming vertices, we may assume $F \subseteq F_1, F_2$. Let $S_1, \ldots, S_\ell$ be the components of F_1, where S_i has order $a_i + 1$, and let $T_1, \ldots, T_\ell$ be the components of F_2, where T_j has order $b_j + 1$. Let H be the bipartite graph with the two partite sets $\{S_1, \ldots, S_\ell\}$ and $\{T_1, \ldots, T_\ell\}$, where S_i is adjacent to T_j if and only if some edge of F belongs to S_i as well as T_j. Since all considered components are stars, the edges of H form a matching M in H. Now, if $S_iT_j \in M$, then the choice of F implies that $\min\{a_i, b_j\}$ edges of S_i and T_j belong to F, that is, $m(F) = \sum\limits_{S_iT_j \in M} \min\{a_i, b_j\}$. In view of this formula, we may assume that M is a perfect matching of H. In other words, there is a permutation π of $[\ell]$ such that $m(F) = \sum\limits_{i=1}^{\ell} \min\{a_i, b_{\pi(i)}\}$. Now, (1) implies that choosing the permutation π as the identity maximizes $\sum\limits_{i=1}^{\ell} \min\{a_i, b_{\pi(i)}\}$, which completes the proof. □

Now, Theorem 1 follows easily by decomposing the given forests into unions of stars. Note that Lemma 1 assumes that F_1 and F_2 have equally many components, which can easily be ensured by adding isolated vertices.

Proof: [Proof of Theorem 1] Let F be a common subgraph of F_1 and F_2 with $m(F) = \mathrm{lcs}(F_1, F_2)$. By renaming vertices, we may assume $F \subseteq F_1, F_2$. For $i \in [2]$, let the set R_i contain exactly one vertex from every component of F_i. Recall that the distance of an edge e from R_i in F_i is the minimum length of a path in F_i intersecting both e and R_i. For $i \in [2]$, let F_i^{even} be the spanning subgraph of F_i containing all edges of F_i that have even distance to R_i, and let $F_i^{\text{odd}} = F_i - E\left(F_i^{\text{even}}\right)$. By construction, all components of F_1^{even}, F_1^{odd}, F_2^{even}, and F_2^{odd} are stars. Furthermore, one of the four sets $E\left(F_1^{\text{even}}\right) \cap E\left(F_2^{\text{even}}\right)$, $E\left(F_1^{\text{even}}\right) \cap E\left(F_2^{\text{odd}}\right)$, $E\left(F_1^{\text{odd}}\right) \cap E\left(F_2^{\text{even}}\right)$, and $E\left(F_1^{\text{odd}}\right) \cap E\left(F_2^{\text{odd}}\right)$ contains at least $1/4$ of the edges of F. Hence, efficiently determining four common subgraphs of maximum sizes for the pairs $(F_1^{\text{even}}, F_2^{\text{even}})$, $(F_1^{\text{even}}, F_2^{\text{odd}})$, $(F_1^{\text{odd}}, F_2^{\text{even}})$, and $(F_1^{\text{odd}}, F_2^{\text{odd}})$ using Lemma 1, and returning the one with most edges, yields a common subgraph of F_1 and F_2 with at least $\frac{1}{4}\mathrm{lcs}(F_1, F_2)$ edges. □

3 Proof of Theorem 2

In this section, we show Theorem 2.

As one ingredient of the proof we need that a largest common subgraph of two given forests with components of bounded orders can be found efficiently by a straightforward dynamic programming approach; the next lemma gives details. For a positive integer Δ, let $\mathcal{F}_\Delta$ be the collection of all forests whose components have orders at most Δ.

Lemma 2. *For every $\Delta \in \mathbb{N}$, there is some $k \in \mathbb{N}$ with the following property: For two given forests F_1 and F_2 of orders at most n from $\mathcal{F}_\Delta$, one can determine a common subgraph F of F_1 and F_2 with $m(F) = \mathrm{lcs}(F_1, F_2)$ in time $O(n^k)$.*

Proof: Let $\Delta \in \mathbb{N}$ be fixed. Let $\{T_1, \ldots, T_p\}$ be the set of all (unrooted) trees of order at most Δ, in particular, p is bounded in terms of Δ. For every forest F of order n from $\mathcal{F}_\Delta$, there is a unique p-tuple $t(F) = (t_1, \ldots, t_p) \in [n]_0^p$ such that F is isomorphic to the disjoint union of t_i copies of T_i for $i \in [p]$. For a forest F of order n from $\mathcal{F}_\Delta$, note that every spanning subforest F' of F also belongs to $\mathcal{F}_\Delta$ and let

$$\mathcal{T}(F) = \{t(F') : \ F' \text{ is a spanning subforest of } F\} \subseteq [n]_0^p.$$

Now, let F be some fixed forest of order at most n from $\mathcal{F}_\Delta$. Let $K_1, \ldots, K_\ell$ be the components of F. For $i \in [\ell]$, let n_i be the order of K_i and let $F_{[i]} = K_1 \cup \cdots \cup K_i$. Since,

$$\begin{array}{rcl} |\mathcal{T}(K_{i+1})| & \leq & (n_{i+1}+1)^p \leq (\Delta+1)^p, \\ |\mathcal{T}(F_{[i]})| & \leq & \left(1+\sum\limits_{j=1}^{i} n_j\right)^p \leq (n+1)^p, \\ F_{[i+1]} & = & F_{[i]} \cup K_{i+1}, \text{ and, hence,} \\ \mathcal{T}(F_{[i+1]}) & = & \left\{t'+t'' : t' \in \mathcal{T}(F_{[i]}) \text{ and } t'' \in \mathcal{T}(K_{i+1})\right\}, \end{array}$$

a simple dynamic programming procedure allows to determine in time $O(n^k)$, for some k depending only on Δ, the set $\mathcal{T}(F)$ and

$$\text{lcs}(F_1,F_2) = \max\left\{\sum_{i=1}^{p} t_i m(T_i) : (t_1,\ldots,t_p) \in \mathcal{T}(F_1) \cap \mathcal{T}(F_2)\right\}.$$

Along the dynamic programming, one can also maintain suitable realizers and the desired statement follows. □

As explained after Theorem 2, we approximate the two given forests by simpler forests that are composed of copies of few different trees that are structurally close to stars. The following two lemmas contain the details.

Let $\epsilon > 0$ and let Δ be a positive integer.

Let

$$\{T_1,\ldots,T_p\} \tag{2}$$

be the set of all rooted trees of order at most Δ, where T_p is the rooted tree of order 1. It is well-known that the number of rooted non-isomorphic trees of order $n+1$ is at most the *n-th Catalan number* $C_n = \frac{1}{n+1}\binom{2n}{n}$ [16]. Since the Catalan numbers are non-decreasing,

$$p \leq \sum_{i=1}^{\Delta} C_{i-1} < \Delta C_\Delta < \binom{2\Delta}{\Delta}.$$

Let

$$D(\epsilon,\Delta) = [\Delta]_0 \cup \left\{\lceil (1+\epsilon)^i \rceil : i \in \mathbb{N}_0\right\}. \tag{3}$$

We say that a forest F is *(ϵ,Δ)-clean* if in each component K of F we can choose and fix a root vertex r_K such that

(i) every component of $K - r_K$ has order at most Δ,

(ii) the degree $d_F(r_K)$ of r_K in F belongs to $D(\epsilon,\Delta)$, and

(iii) for every rooted tree T in $\{T_1,\ldots,T_{p-1}\}$, that is, the order of T is at least 2, the number of components L of $K - r_K$, considered as trees rooted in the neighbor of r_K in $V(L)$, that are isomorphic to T as a rooted tree is a multiple of

$$\max\left\{1, \left\lfloor \frac{\epsilon d_F(r_K)}{\Delta\binom{2\Delta}{\Delta}} \right\rfloor\right\}. \tag{4}$$

See Figure 1 for an illustration.

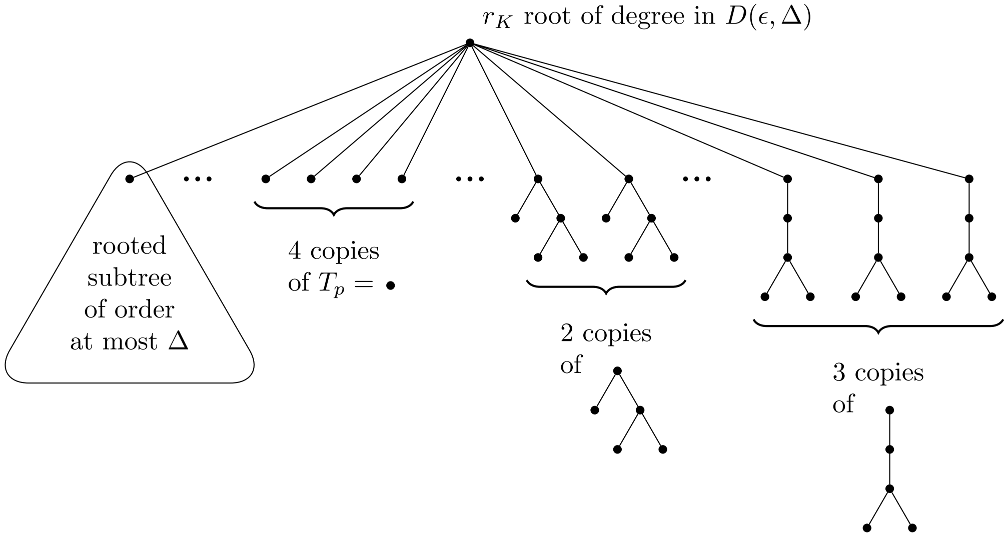
r_K root of degree in $D(\epsilon,\Delta)$
rooted subtree of order at most Δ
4 copies of $T_p = \bullet$
2 copies of
3 copies of

Figure 1: A component K with root r_K of an (ϵ,Δ)-clean forest F. Note that we consider the components L of $K - r_K$ as trees rooted in the neighbor of r_K in $V(L)$, which means that we distinguish isomorphic components of $K - r_K$ that are attached differently to the root r_K. The figure shows five components of $K - r_K$ that are all isomorphic as unrooted trees to the unique tree with degree sequence $3, 2, 1, 1, 1$. Two of these components are attached to r_k at their vertex of degree 2 and three are attached to r_k differently.

Lemma 3. *For every $\epsilon \in (0,1)$ and $\Delta \in \mathbb{N}$ with $\epsilon\Delta \geq 1$, there is some $k \in \mathbb{N}$ with the following property: For a given forest F of order n, one can determine a spanning (ϵ,Δ)-clean subforest F' of F with*

$$m(F') \geq \left(1 - 4\left(\epsilon + \frac{1}{\Delta}\right)\right) m(F)$$

in time $O(n^k)$.

Proof: Let F be a forest of size m. Since the degree sum of F is $2m$, there are less than $\frac{2m}{\Delta}$ vertices in F that have degree more than Δ. Rooting every component of F in some vertex and choosing, for every vertex of degree more than Δ that is no root, the edge to its parent, yields a set E_0 of less than $\frac{2m}{\Delta}$ edges of F such that every component of $F_0 = F - E_0$ contains at most one vertex of degree more than Δ. We are now going to ensure the three properties (i), (ii), and (iii) from the definition of (ϵ,Δ)-cleanness by removing further edges in three consecutive cleaning steps.

For every component K of F_0, choose a vertex r_K of maximum degree within K as its root. Call a component K $\frac{1}{3}$-*clean* if every component of $K - r_K$ has order at most Δ. Suppose that K is a component that is not $\frac{1}{3}$-clean. Let u be some vertex of maximum depth in K (rooted in r_K) such that one plus the number of descendants of u in K is more than Δ. Removing the edge between u and its parent cuts off a component L containing u that has at least Δ edges. Choosing u as its root r_L, the component L is $\frac{1}{3}$-clean. For the remaining part $K - V(L)$ of K, we keep r_K as its root. Iteratively repeating this procedure as long as there are components that are not $\frac{1}{3}$-clean, yields a set E_1 of at most $\frac{m(F_0)}{\Delta} \leq \frac{m}{\Delta}$ edges of F_0 such that every component K of $F_1 = F_0 - E_1$ has a specified root r_K and is $\frac{1}{3}$-clean.

For a component K of F_1, let d_K denote the degree of r_K in F_1. Call a component K $\frac{2}{3}$-*clean* if it is $\frac{1}{3}$-clean and d_K belongs to $D(\epsilon, \Delta)$. For $i \in \mathbb{N}_0$, let $d_i = \lceil (1+\epsilon)^i \rceil$, that is, $D(\epsilon, \Delta) = [\Delta]_0 \cup \{d_i : i \in \mathbb{N}_0\}$. For every component K of F_1 with $d_K \notin D(\epsilon, \Delta)$, let i_K be the largest non-negative integer i with $d_i \leq d_K$, in particular,

$$(1+\epsilon)^{i_K} \leq d_{i_K} \leq d_K < d_{i_K+1} < (1+\epsilon)^{i_K+1} + 1.$$

Since $d_K \notin D(\epsilon, \Delta)$, we have $d_K - 1 \geq \Delta$. This implies $1 \leq \epsilon\Delta \leq \epsilon(d_K - 1) \leq \epsilon(1+\epsilon)^{i_K+1}$, and, hence,

$$d_K < (1+\epsilon)^{i_K+1} + 1 \leq (1+\epsilon)^{i_K+1} + \epsilon(1+\epsilon)^{i_K+1} = (1+\epsilon)^{i_K+2}.$$

Let the set E_2 of edges of F_1 contain exactly $d_K - d_{i_K}$ many edges incident with r_K for every component K of F_1 with $d_K \notin D(\epsilon, \Delta)$. Since

$$\frac{d_K - d_{i_K}}{d_K} \leq \frac{(1+\epsilon)^{i_K+2} - (1+\epsilon)^{i_K}}{(1+\epsilon)^{i_K}} = 2\epsilon + \epsilon^2 \leq 3\epsilon,$$

the set E_2 contains at most a 3ϵ-fraction of the edges of F_1, that is, $|E_2| \leq 3\epsilon m(F_1) \leq 3\epsilon m$. Let $F_2 = F_1 - E_2$. For every component K of F_2 that contains the root r of some component of F_1, choose r as the root r_K of K. Each component K of F_2 that does not contain the root of some component of F_1 has order at most Δ, and we choose any of its vertices as its root r_K. With these choices of the roots, each component of F_2 is $\frac{2}{3}$-clean.

Call a component K of F_2 *clean* if it is $\frac{2}{3}$-clean and, for every rooted tree T of order at least 2, the number of components L of $K - r_K$, considered as trees rooted in the neighbor of r_K in $V(L)$, that are isomorphic to T as a rooted tree is a multiple of

$$\delta = \max\left\{1, \left\lfloor \frac{\epsilon d_{F_2}(r_K)}{\Delta\binom{2\Delta}{\Delta}} \right\rfloor\right\}.$$

Suppose that some component K of F_2 is not clean. This implies that $\delta = \left\lfloor \frac{\epsilon d_{F_2}(r_K)}{\Delta\binom{2\Delta}{\Delta}} \right\rfloor > 1$. For every component L of $K - r_K$, choose the neighbor of r_K in $V(L)$ as its root. Let $\{T_1, \ldots, T_p\}$ be as in (2). Let $K - r_K$ contain exactly t_i components that are isomorphic to T_i as rooted trees for every $i \in [p]$. Now, for every $i \in [p-1]$, remove all edges of exactly $t_i - \left\lfloor \frac{t_i}{\delta} \right\rfloor \delta$ copies of the rooted tree T_i among the components of $K - r_K$. This does not affect the degree of r_K and results in a subforest K' of K in which the component containing the root r_K is clean and all remaining components are isolated vertices, which means that they are also clean. Furthermore, since each T_i has at most $\Delta - 1$ edges, we obtain

$$|E(K) \setminus E(K')| \quad = \quad m(K) - m(K') \leq (p-1)(\Delta-1)\delta < \binom{2\Delta}{\Delta}\Delta\delta \leq \epsilon d_{F_2}(r_K) \leq \epsilon m(K).$$

Performing this operation for every component of F_2 that is not clean, yields a forest $F' = F_2 - E_3$ that is (ϵ, Δ)-clean. The set E_3 of removed edges satisfies $|E_3| \leq \epsilon m(F_2) \leq \epsilon m$.

Now,

$$m(F') = m - |E_0| - |E_1| - |E_2| - |E_3| \geq m - \left(\frac{2}{\Delta} + \frac{1}{\Delta} + 3\epsilon + \epsilon\right) m \geq m - 4\left(\epsilon + \frac{1}{\Delta}\right) m.$$

Furthermore, all steps of the cleaning procedures can be performed in polynomial time for fixed ϵ and Δ. This completes the proof. □

A natural choice for Δ is $\lceil \frac{1}{\epsilon} \rceil$, which immediately implies $\epsilon\Delta \geq 1$.

Accordingly, let a forest be *ϵ-clean* if it is $\left(\epsilon, \lceil \frac{1}{\epsilon} \rceil\right)$-clean and let

$$\begin{aligned} \mathcal{T}(\epsilon) &= \{T_1, \dots, T_p\} \text{ be as in (2) for } \Delta = \lceil \tfrac{1}{\epsilon} \rceil \text{ as well as} && (5) \\ D(\epsilon) &= D(\epsilon, \Delta) \text{ be as in (3) for } \Delta = \lceil \tfrac{1}{\epsilon} \rceil. && (6) \end{aligned}$$

Lemma 4. *For every $\epsilon \in (0, 1)$, there are $c_1, c_2, k \in \mathbb{N}$ with the following property: For every positive integer n at least 2, there is a set $\mathcal{S}(\epsilon, n)$ of at most $c_1 \log(n)$ rooted trees such that every component of every ϵ-clean forest of order n belongs to $\mathcal{S}(\epsilon, n)$. Furthermore, if $S_1, \dots, S_q$ is a linear ordering of the elements of $\mathcal{S}(\epsilon, n)$ such that the degrees d_i of the roots r_i of S_i are non-decreasing along this ordering, and $i, j \in [q]$ are such that $j \geq i + c_2$, then $d_i \leq \epsilon d_j$. Finally, $\mathcal{S}(\epsilon, n)$ can be constructed in time $O(n^k)$.*

Proof: Let $\Delta = \lceil \frac{1}{\epsilon} \rceil$. Let F be an ϵ-clean forest of order n. Let K be a component of F with root vertex r_K.

If $d_F(r_K) < \frac{\Delta}{\epsilon}\binom{2\Delta}{\Delta}$, then

$$n(K) \leq 1 + d_F(r_K)\Delta \leq 1 + \frac{\Delta^2}{\epsilon}\binom{2\Delta}{\Delta},$$

which implies that, for fixed ϵ, there are constantly many choices for such a component.

Now, let $d_F(r_K) \geq \frac{\Delta}{\epsilon}\binom{2\Delta}{\Delta}$. For δ as in (4), we obtain

$$\delta = \max\left\{1, \left\lfloor \frac{\epsilon d_F(r_K)}{\Delta\binom{2\Delta}{\Delta}} \right\rfloor\right\} = \left\lfloor \frac{\epsilon d_F(r_K)}{\Delta\binom{2\Delta}{\Delta}} \right\rfloor \geq \frac{\epsilon d_F(r_K)}{2\Delta\binom{2\Delta}{\Delta}}.$$

As in (5), let $\{T_1, \dots, T_p\}$ be the set of all rooted trees of order at most Δ, where T_p is the rooted tree of order 1. Let $K - r_K$ contain exactly t_i components that are isomorphic to T_i as rooted trees for every $i \in [p]$. Since $t_i \leq d_F(r_K)$ for every $i \in [p-1]$, property (iii) in the definition of (ϵ, Δ)-cleanness implies that there are at most

$$1 + \frac{d_F(r_K)}{\delta} \leq 1 + \frac{2\Delta\binom{2\Delta}{\Delta}}{\epsilon}$$

possible values for each t_i with $i \in [p-1]$. Recall that T_p is the rooted tree of order 1 and that

$$t_p = d_F(r_K) - (t_1 + \dots + t_{p-1}),$$

which implies that K is determined up to isomorphism by $t_1, \dots, t_{p-1}$ and the degree of its root. Therefore, for every fixed integer d with $\frac{\Delta}{\epsilon}\binom{2\Delta}{\Delta} \leq d \leq n$, there are at most $c_3 := \left(1 + \frac{2\Delta\binom{2\Delta}{\Delta}}{\epsilon}\right)^{p-1}$ many choices for K such that the degree $d_F(r_K)$ of its root r_K equals d. Recall that p is bounded in terms of Δ, which, in turn, is bounded in terms of ϵ. Since $D(\epsilon)$ as in (6) contains at most $\log_{(1+\epsilon)}(n) = \frac{\log(n)}{\log(1+\epsilon)}$ such values d, there is some integer c_1 depending only on ϵ, and there is a set $\mathcal{S}(\epsilon, n)$ of at most $c_1 \log(n)$ rooted trees such that every component of every ϵ-clean forest of order n belongs to $\mathcal{S}(\epsilon, n)$.

Now, let

$$S_1, \ldots, S_q$$

be a linear ordering of the elements of $\mathcal{S}(\epsilon, n)$ such that the degrees d_i of the roots r_i of S_i are non-decreasing along this ordering. This ordering begins with constantly many rooted trees with roots of degrees d_i at most $\frac{\Delta}{\epsilon}\binom{2\Delta}{\Delta}$. Once the root degrees d_i are at least this value, the structure of $D(\epsilon)$ implies that they increase by a factor of $(1+\epsilon)$ after every $O(c_3)$ steps in the ordering. This implies the existence of some positive integer c_2 such that, for every $i, j \in [q]$ with $j \geq i + c_2$, we have $d_i \leq \epsilon d_j$.

The above arguments imply that $\mathcal{S}(\epsilon, n)$ can be constructed in time $O(n^k)$ for some integer k depending only on ϵ. □

We are now in a position to complete the proof.

Proof: [Proof of Theorem 2] Let $\epsilon \in (0,1)$ be fixed. Let $\Delta = \left\lceil \frac{1}{\epsilon} \right\rceil$. Within this proof we call a forest *clean* if it is $\left(\epsilon, \left\lceil \frac{1}{\epsilon} \right\rceil\right)$-clean. Let $\mathcal{T} = \mathcal{T}(\epsilon)$ be as in (6), that is,

$$\mathcal{T} = \{T_1, \ldots, T_p\}$$

is the set of all rooted trees of order at most Δ, where T_p is the rooted tree of order 1. Let $\mathcal{D} = \mathcal{D}(\epsilon)$ be as in (6), that is, $\mathcal{D} = [\Delta]_0 \cup \left\{ \left\lceil (1+\epsilon)^i \right\rceil : i \in \mathbb{N}_0 \right\}$.

Now, let F_1 and F_2 be two given forests of order n at least 2, for which we want to determine a common subgraph F of large size. Note that, in view of the desired statement, it would suffice that $m(F) \geq \mathrm{lcs}(F_1, F_2) - C\epsilon n$ for some constant C independent of ϵ and n.

Cleaning the given forests

Suppose that F_1 or F_2 are not clean. Using Lemma 3, we can determine in polynomial time a set E_1 of edges of F_1 and a set E_2 of edges of F_2 such that $F_1' = F_1 - E_1$ and $F_2' = F_2 - E_2$ are clean and

$$|E_1| + |E_2| \leq 4\left(\epsilon + \frac{1}{\Delta}\right)(m(F_1) + m(F_2)) \leq 4(\epsilon + \epsilon)\, 2n = 16\epsilon n.$$

If F is a common subgraph of F_1 and F_2, then removing from F the at most $16\epsilon n$ edges corresponding to edges from E_1 or E_2 that belong to F yields a common subgraph F' of F_1' and F_2' such that $m(F') \geq m(F) - 16\epsilon n$, in particular, $\mathrm{lcs}(F_1', F_2') \geq \mathrm{lcs}(F_1, F_2) - 16\epsilon n$. In view of the desired statement, we may therefore assume that

$$F_1 \text{ and } F_2 \text{ are clean.}$$

Using Lemma 4, we construct in polynomial time the set

$$\mathcal{S} = \mathcal{S}(\epsilon, n) = \{S_1, \ldots, S_q\}$$

and the integers c_1 and c_2 as in Lemma 4, that is, $\mathcal{S}$ contains $q \leq c_1 \log(n)$ clean rooted trees and every component of F_1 and F_2 belongs to $\mathcal{S}$. Furthermore, denoting the root of S_i and its degree by r_i and d_i, respectively, we have

$$d_i \leq \epsilon d_j \text{ for every } i, j \in [q] \text{ with } j \geq i + c_2. \tag{7}$$

Notational interlude

Let F be a common subgraph of F_1 and F_2. Extending an isomorphism between a subgraph of F_1 that is isomorphic to F and a subgraph of F_2 that is also isomorphic to F yields a bijection $f : V(F_1) \to V(F_2)$ with the following property: F is isomorphic to a subgraph of the forest F_f, where F_f has vertex set $V(F_1)$ and contains all edges uv of F_1 for which $f(u)f(v)$ is an edge in F_2. In fact, F_f itself is a spanning common subgraph of F_1 and F_2, and

$$\text{lcs}(F_1, F_2) = \max\{m(F_f) : f : V(F_1) \to V(F_2) \text{ bijective}\}.$$

Possibly after adding isolated vertices and renaming vertices, we may assume now and later, for notational convenience, that F is a spanning subgraph of F_f. See Figure 2 for an illustration.

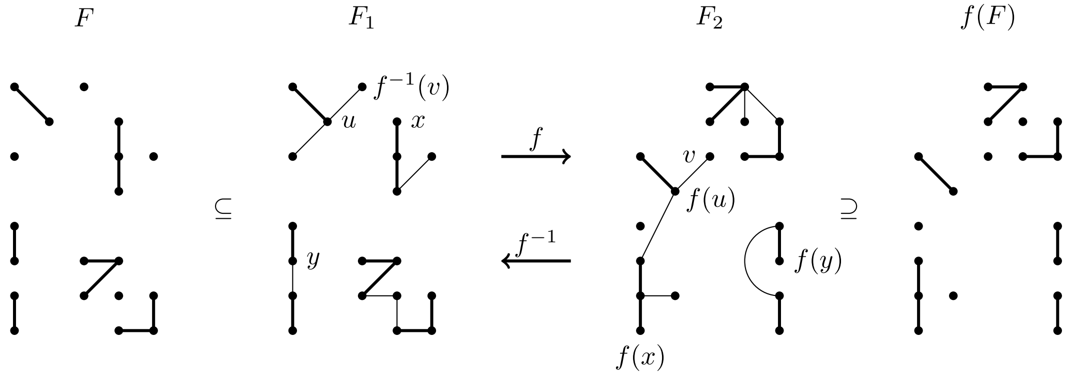

Figure 2: A common subgraph F of F_1 and F_2 as a spanning subgraph of F_1 together with a corresponding bijection $f : V(F_1) \to V(F_2)$. The forest $f(F)$ with vertex set $V(F_2)$ and edge set $\{f(x)f(y) : xy \in E(F)\}$ is a spanning subgraph of F_2. Note that F_f contains strictly more edges than F; the edge $uf^{-1}(v)$ of F_1 belongs to F_f, because the edge $f(u)v$ belongs to F_2. Up to isomorphism, F is described by the multiplicities of its components; it consists of 4 copies of T_p, 3 copies of K_2, and 3 copies of P_3. Note that there are two non-isomorphic ways to choose a root for P_3.

Nice solutions — pairing or isolating large degree roots

We call a common subgraph F of F_1 and F_2 *nice* if there is some bijection $f : V(F_1) \to V(F_2)$ such that F is a spanning subgaph of F_f and F is a common subgraph of the two forests F_1' and F_2' constructed as follows:

- F_1' is the spanning subgraph of F_1 that is obtained by removing all edges incident with every root vertex r of some component of F_1 such that $d_{F_1}(r) \geq \frac{\Delta}{\epsilon}$ and either $\epsilon d_{F_1}(r) \geq d_{F_2}(f(r))$ or $\epsilon d_{F_2}(f(r)) \geq d_{F_1}(r)$. Note that, in both cases,
$$\min\{d_{F_1}(r), d_{F_2}(f(r))\} \leq \epsilon \max\{d_{F_1}(r), d_{F_2}(f(r))\} \leq \epsilon(d_{F_1}(r) + d_{F_2}(f(r))). \quad (8)$$
- F_2' is the spanning subgraph of F_2 that is obtained by removing all edges incident with every root vertex r of some component of F_2 such that $d_{F_2}(r) \geq \frac{\Delta}{\epsilon}$ and either $\epsilon d_{F_2}(r) \geq d_{F_1}(f^{-1}(r))$ or $\epsilon d_{F_1}(f^{-1}(r)) \geq d_{F_2}(r)$. Again, in both cases,
$$\min\{d_{F_1}(f^{-1}(r)), d_{F_2}(r)\} \leq \epsilon \max\{d_{F_1}(f^{-1}(r)), d_{F_2}(r)\} \leq \epsilon(d_{F_1}(f^{-1}(r)) + d_{F_2}(r)).$$

Note that if a vertex r of F_1 is of degree at least $\frac{\Delta}{\epsilon}$, then it is necessarily the root of some component of F_1. Hence, if a vertex r of F_1 is of degree at least $\frac{\Delta}{\epsilon}$ and $f(r)$ is no root of some component of F_2, then the degree of $f(r)$ in F_2 is at most Δ. It follows that $\epsilon d_{F_1}(r) \geq d_{F_2}(f(r))$, which implies that r will be isolated in the nice subgraph F.

Now, suppose that F is a common subgraph of F_1 and F_2 with $m(F) \geq \mathrm{lcs}(F_1, F_2) - \epsilon n$ that is not nice. For convenience, we may assume that F is a spanning subgraph of F_f for some bijection f. Since F is not nice, F is not a common subgraph of F_1' and F_2' as defined above using this f. Since the degree of a root r of some component of F_1 within the forest F is at most $\min\{d_{F_1}(r), d_{F_2}(f(r))\}$, (8) implies that removing at most

$$\begin{aligned}
\sum_{r\in V(F_1)} \min\{d_{F_1}(r), d_{F_2}(f(r))\} &\leq \epsilon \sum_{r\in V(F_1)} (d_{F_1}(r) + d_{F_2}(f(r))) \\
&= \epsilon \left(\sum_{r\in V(F_1)} d_{F_1}(r) + \sum_{r'\in V(F_2)} d_{F_2}(r') \right) \\
&= 2\epsilon(m(F_1) + m(F_2)) \leq 4\epsilon n
\end{aligned}$$

edges from F yields a subgraph F' of F_1'. Similarly, removing at most $4\epsilon n$ further edges from this subgraph F' yields a subgraph F'' of F_1' and F_2'. Now, F'' is a nice common subgraph of F_1 and F_2, and

$$m(F'') \geq m(F) - 8\epsilon n \geq \mathrm{lcs}(F_1, F_2) - 9\epsilon n.$$

Therefore, in view of the desired statement, it suffices to determine in polynomial time a nice common subgraph F of F_1 and F_2 that is a spanning subgraph of F_1 and has sufficiently many edges.

(Potentially) large components in nice common subgraphs

Our approach to find a sufficiently good nice common subgraph of F_1 and F_2 consists in efficiently generating polynomially many options for the roles of the high degree root vertices of components of F_1 and F_2 within components of the common subgraph that are (potentially) of order at least $1 + \frac{\Delta^2}{\epsilon}$. Removing the corresponding parts from F_1 and F_2 yields forests F_1' and F_2', whose components all have orders less than $1 + \frac{\Delta^2}{\epsilon}$, and Lemma 2 allows to determine common subgraphs of F_1' and F_2' of maximum size in polynomial time. Returning the best overall solution encountered in this way, while considering the polynomially many options for the high degree roots, yields a sufficiently good nice common subgraph of F_1 and F_2.

Let F be a nice common subgraph of F_1 and F_2 that is a spanning subgraph of the forest F_f for some bijection $f : V(F_1) \to V(F_2)$. Let K be a component of F that contains a vertex r with $d_{F_1}(r) \geq \frac{\Delta}{\epsilon}$ or $d_{F_2}(f(r)) \geq \frac{\Delta}{\epsilon}$. Since F_1 is clean, every component of F that does not contain such a vertex necessarily has order less than $1 + \frac{\Delta^2}{\epsilon}$.

It follows that

- K is an induced subgraph of the component K_1 of F_1 with root r and
- the tree $f(K)$ with vertex set $f(V(K))$ and edge set $\{f(u)f(v) : uv \in E(K)\}$ is an induced subgraph of the component K_2 of F_2 with root $f(r)$.

Since F_1 and F_2 are clean, all their components are rooted trees from $\mathcal{S}$. Let K_1 be a copy of S_{i_1} and let K_2 be a copy of S_{i_2}. Denote the root r of K_1 by r_{i_1} and the root $f(r)$ of K_2 by r_{i_2}.

For every child u of r_{i_1} in K_1,

- either the edge $r_{i_1}u$ does not belong to F, which means that u does not belong to K,
- or there is some child v of r_{i_2} in K_2 such that $r_{i_1}u$ belongs to $K \subseteq F$, u belongs to K, $f(u) = v$, and v belongs to $f(K)$.

Symmetric options hold for every child v of r_{i_2} in K_2.

If some child u of r_{i_1} in K_1 belongs to K and the child $v = f(u)$ of r_{i_2} in K_2 belongs to $f(K)$, then the component of $K_1 - r_{i_1}$ that contains u is (the copy of) a tree T_{j_1} from $\mathcal{T}$ rooted in u, and the component of $K_2 - r_{i_2}$ that contains v is (the copy of) a tree T_{j_2} from $\mathcal{T}$ rooted in v. We now consider the options for the subtrees of K within $K_1 - r_{i_1}$ and of $f(K)$ within $K_2 - r_{i_2}$.

The constantly many options to overlay trees from $\mathcal{T}$ at their roots

Let X be the set of all 5-tuples x such that

- either $x = (j_1, j_2, A_0, A_1, A_2)$, where
  - $j_1, j_2 \in [p]$,
  - A_0 is one of the rooted trees from $\{T_1, \dots, T_p\}$,
  - for each $\ell \in [2]$, A_0 is isomorphic as a rooted tree to a rooted subtree $A_{0,\ell}$ of T_{j_ℓ} that is rooted in the root of T_{j_ℓ},
  - $A_1 \cong T_{j_1} - V(A_{0,1})$, and
  - $A_2 \cong T_{j_2} - V(A_{0,2})$,

  see Figure 3 for an illustration,
- or $x = (j_1, \emptyset, \emptyset, \emptyset, \emptyset)$ for $j_1 \in [p]$, corresponding to the option that the child of r_{i_1} in K_1 that belongs to a copy of T_{j_1} in $K_1 - r_{i_1}$ does not belong to K.
- or $x = (\emptyset, j_2, \emptyset, \emptyset, \emptyset)$ for $j_2 \in [p]$, corresponding to the option that the child of r_{i_2} in K_2 that belongs to a copy of T_{j_2} in $K_2 - r_{i_2}$ does not belong to $f(K)$.

Note that we add the 5-tuples of the form $(j_1, \emptyset, \emptyset, \emptyset, \emptyset)$ and $(\emptyset, j_2, \emptyset, \emptyset, \emptyset)$ for notational convenience: Together with the 5-tuples of the form $(j_1, j_2, A_0, A_1, A_2)$, they allow to clarify the role of all children (up to symmetry) of the root of K_1 as well as of all children of the root of K_2 within K.

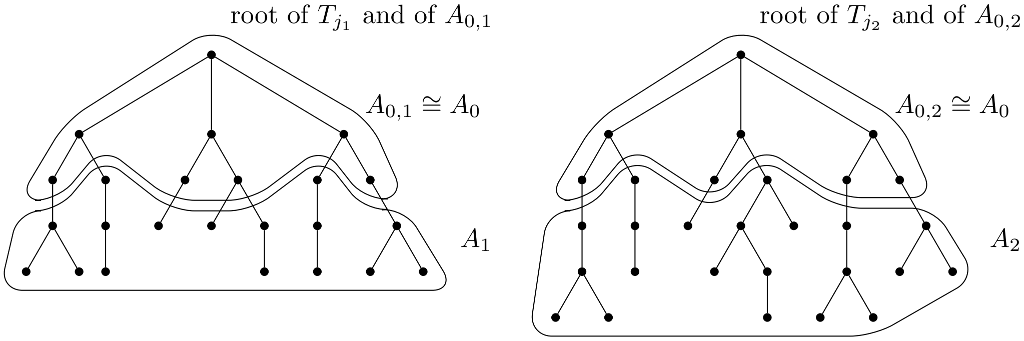
root of T_{j_1} and of $A_{0,1}$
$A_{0,1} \cong A_0$
A_1
root of T_{j_2} and of $A_{0,2}$
$A_{0,2} \cong A_0$
A_2

Figure 3: Subgraphs $A_{0,1}$ and A_1 of $T_{j_1} \subseteq F_1$ (on the left) and $A_{0,2}$ and A_2 of $T_{j_2} \subseteq F_2$ (on the right) for some $x = (j_1, j_2, A_0, A_1, A_2)$ in X. Note that there are different isomorphic copies of A_0 within T_{j_1} and T_{j_2} containing their roots, which lead to different possibilities for the subforests A_1 and A_2, completely specified up to isomorphism by the considered 5-tuples.

Since, for fixed ϵ, the number p of trees in $\mathcal{T}$ is bounded by a constant and all these trees have order at most Δ, the set X is bounded by a constant for fixed ϵ. Let

$$X = \{x_1, \dots, x_o\}.$$

For $j \in [p]$, let $X_{(j,*,*,*,*)}$ be the subset of all 5-tuples in X that have j as their first entry, and let $X_{(*,j,*,*,*)}$ be the subset of all 5-tuples in X that have j as their second entry.

For each $\ell \in [2]$ and $j \in [p]$, let $S_{i_\ell} - r_{i_\ell}$ contain $t_{\ell,j}$ components of T_j, that is, the p-tuples $(t_{1,1}, \dots, t_{1,p})$ and $(t_{2,1}, \dots, t_{2,p})$ determine S_{i_1} and S_{i_2} up to isomorphism, respectively. In particular, $t_{\ell,1} + \dots + t_{\ell,p}$ equals the degree d_{i_ℓ} of the root r_{i_ℓ} of S_{i_ℓ}. See Figure 1 illustrating the structure of the rooted trees in $\mathcal{S}$.

Quantizing the options for a single (possibly) large component

All essentially different options for K within $K_1 \cong S_{i_1}$ and $f(K)$ within $K_2 \cong S_{i_2}$ can be encoded in a natural way by an o-tuple $(y_1, \dots, y_o) \in \mathbb{N}_0^o$; in particular, for $x_\ell = (j_1, j_2, A_0, A_1, A_2)$ in X, there are y_ℓ pairs (L_1, L_2) such that L_1 is a component of $K_1 - r_{i_1}$ isomorphic to T_{j_1}, L_2 is a component of $K_2 - r_{i_2}$ isomorphic to T_{j_2}, K contains the root of L_1, $f(K)$ contains the root of L_2, $K \cap L_1 \cong A_0$, $f(K) \cap L_2 \cong A_0$, $L_1 - V(K) \cong A_1$, and $L_2 - V(f(K)) \cong A_2$. Note that

$$t_{1,j} = \sum_{x_\nu \in X_{(j,*,*,*,*)}} y_\nu \quad \text{and} \quad t_{2,j} = \sum_{x_\nu \in X_{(*,j,*,*,*)}} y_\nu \quad \text{for every } j \in [p]. \tag{9}$$

Let

$$Y_{(i_1,i_2)}$$

denote the set of all these o-tuples, which depends only on $i_1, i_2 \in [q]$. In principle, all o-tuples in $Y_{(i_1,i_2)}$ may be relevant for optimally solving the largest common subgraph problem for F_1 and F_2. Their total number though would lead to a superpolynomial running time. Since we aim for an approximate solution only, we may restrict these options by quantization. Therefore, let $\tilde{Y}_{(i_1,i_2)}$ be the set of all o-tuples $(y_1, \dots, y_o)$ in $Y_{(i_1,i_2)}$ such that, for every $\nu \in [o]$ such that the first two entries of x_ν belong to $[p]$, the value of y_ν is a multiple of

$$\delta = \max\left\{1, \left\lfloor \frac{\epsilon(d_{i_1} + d_{i_2})}{o\Delta} \right\rfloor \right\}.$$

Note that, by (9), the y_ν with x_ν of the form $(j, \emptyset, \emptyset, \emptyset, \emptyset)$ or $(\emptyset, j, \emptyset, \emptyset, \emptyset)$ for some $j \in [p]$, are determined by the remaining y_ν and the $t_{\ell,j}$. Since $y_1 + \dots + y_o \le d_{i_1} + d_{i_2}$, the step-size δ leaves at most $1 + \frac{2o\Delta}{\epsilon}$ possible values for each y_ν with $\nu \in [o]$ such that the first two entries of x_ν belong to $[p]$ and, hence,

$$\left|\tilde{Y}_{(i_1,i_2)}\right| \le o_{\max} := \left(1 + \frac{2o\Delta}{\epsilon}\right)^{o-2p},$$

which is constant for fixed ϵ.

Enumerate the elements of this set as

$$\tilde{Y}_{(i_1,i_2)} = \left\{ y^1_{(i_1,i_2)}, \dots, y^{o_{(i_1,i_2)}}_{(i_1,i_2)} \right\}.$$

Restricting, for all $(i_1, i_2) \in [q]^2$, to $\tilde{Y}_{(i_1,i_2)}$ instead of $Y_{(i_1,i_2)}$ for the approximate solution of the largest common subgraph problem for F_1 and F_2, deteriorates the achievable solutions by at most

$2\epsilon n$. In fact, if $\delta = 1$ for $(i_1, i_2) \in [p]^2$, then $Y_{(i_1,i_2)} = \tilde{Y}_{(i_1,i_2)}$ and nothing changes. If $\delta > 1$ for $(i_1, i_2) \in [p]^2$, then, for each o-tuple $(y_1, \dots, y_o)$ in $Y_{(i_1,i_2)} \setminus \tilde{Y}_{(i_1,i_2)}$, reducing each entry y_ν such that the first two entries x_ν belong to $[p]$, say $x_\nu = (j_1, j_2, A_0, A_1, A_2)$, by less than δ, and increasing both entries for $(j_1, \emptyset, \emptyset, \emptyset, \emptyset)$ and $(\emptyset, j_2, \emptyset, \emptyset, \emptyset)$ by exactly the same amount, yields an o-tuple in $\tilde{Y}_{(i_1,i_2)}$. Since each tree in $\mathcal{T}$ has order at most Δ, using this new o-tuple instead of the old one, reduces the number of edges from $K_1 \cong S_{i_1}$ in the solution by at most $\delta o \Delta \le \epsilon(d_{i_1} + d_{i_2})$, which is at most an ϵ-fraction of the number of edges in $K_1 \cong S_{i_1}$ plus the the number of edges in $K_2 \cong S_{i_2}$; this relative local error sums up to at most $2\epsilon n$.

Note that not all elements of $\tilde{Y}_{(i_1,i_2)}$ lead to a component in the solution that contains the roots of copies of S_{i_1} (and S_{i_2}) and has order at least $1 + \frac{\Delta^2}{\epsilon}$, but every such large component corresponds to an element of $\tilde{Y}_{(i_1,i_2)}$.

Quantizing the options for all (possibly) large components

Note that $\mathcal{T}$, $\mathcal{S}$, X, and $\tilde{Y}_{(i_1,i_2)}$ for $i_1, i_2 \in [p]$ only depend on ϵ and n but not on the specific clean forests F_1 and F_2. Having understood and restricted the possible large components arising from a pair of components, one from F_1 and one from F_2, we now consider F_1 and F_2 as a whole.

For $\ell \in [2]$ and $i \in [q]$, let F_ℓ contain $s_{\ell,i}$ components that are copies of S_i, that is,

$$F_1 \cong \bigcup_{i=1}^{q} s_{1,i} S_i \quad \text{and} \quad F_2 \cong \bigcup_{i=1}^{q} s_{2,i} S_i.$$

Consider a nice common subgraph F of F_1 and F_2 that is a spanning subgraph of the forest F_f for some bijection $f : V(F_1) \to V(F_2)$, such that, for every $(i_1, i_2) \in [q]^2$, every component K_1 of F_1 isomorphic to S_{i_1}, and every component K_2 of F_2 isomorphic to S_{i_2} such that f maps the root r of K_1 to the root of K_2, and r is not isolated in F, the common subgraph F is compatible on K_1 and K_2 with some element of $\tilde{Y}_{(i_1,i_2)}$. Note that, in this case, since r is not isolated, (7) and the niceness of F imply $|i_2 - i_1| \le c_2$. For every $i_1 \in [q]$, every $i_2 \in [q]$ with $|i_2 - i_1| \le c_2$, and every $k \in \left[o_{(i_1,i_2)}\right]$, let $s(i_1, i_2, k)$ be the number of components K_1 of F_1 isomorphic to S_{i_1} whose root r is mapped by f to the root of some component K_2 of F_2 isomorphic to S_{i_2} such that r is not isolated in F, and the component K of F that contains r corresponds to the element $y^k_{(i_1,i_2)}$ of $\tilde{Y}_{(i_1,i_2)}$. Note that S_{i_1} has at most $d_{i_1}\Delta$ edges. Reducing each value $s(i_1, i_2, k)$ by less than

$$\delta'(i_1) = \max\left\{1, \left\lfloor \frac{\epsilon s_{1,i_1}}{(2c_2 + 1) o_{\max} \Delta} \right\rfloor \right\}$$

corresponds to isolating certain roots of components of F_1 within F and deteriorates the corresponding overall solution by less than

$$\begin{aligned}
\sum_{i_1 \in [q]} \sum_{i_2 \in [q] : |i_2 - i_1| \le c_2} \frac{\epsilon s_{1,i_1}}{(2c_2+1) o_{\max} \Delta} o_{(i_1,i_2)} d_{i_1} \Delta &\le \sum_{i_1 \in [q]} \sum_{i_2 \in [q] : |i_2 - i_1| \le c_2} \frac{\epsilon s_{1,i_1}}{(2c_2+1)} d_{i_1} \\
&\le \sum_{i_1 \in [q]} \epsilon s_{1,i_1} d_{i_1} \\
&\le \epsilon m(F_1) \\
&\le \epsilon n.
\end{aligned}$$

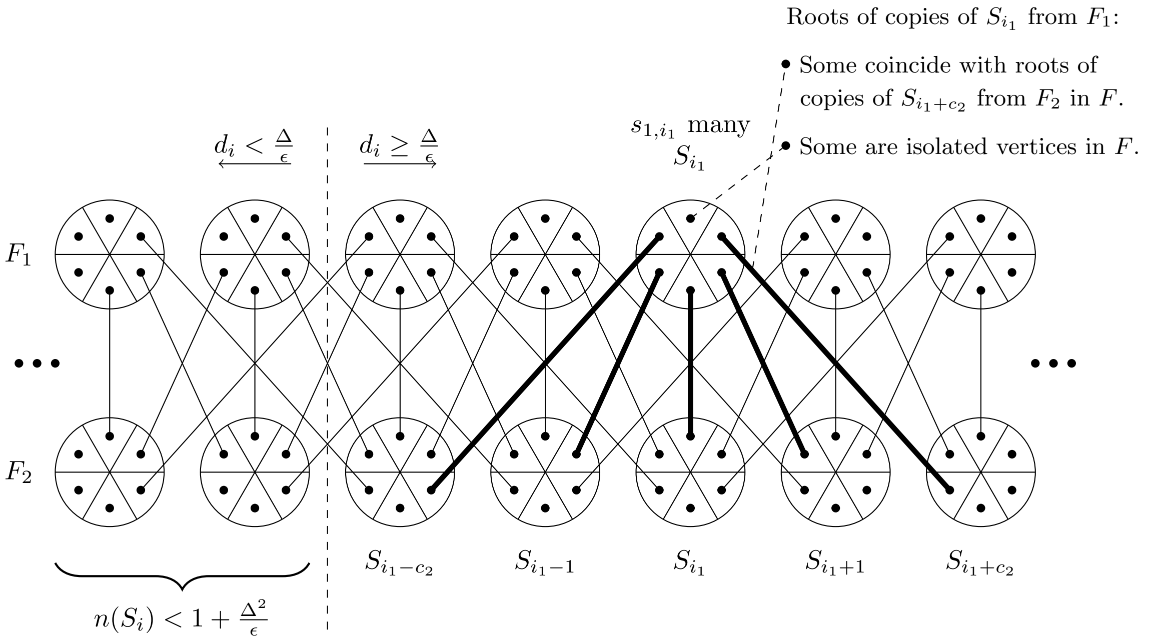
Roots of copies of S_{i_1} from F_1:
Some coincide with roots of copies of $S_{i_1+c_2}$ from F_2 in F.
Some are isolated vertices in F.
s_{1,i_1} many S_{i_1}
$d_i < \frac{\Delta}{\epsilon}$
$d_i \geq \frac{\Delta}{\epsilon}$
F_1
F_2
$S_{i_1-c_2}$
S_{i_1-1}
S_{i_1}
S_{i_1+1}
$S_{i_1+c_2}$
$n(S_i) < 1 + \frac{\Delta^2}{\epsilon}$

Figure 4: The figure illustrates part of the information encoded by $M = \Big(s(i_1, i_2, k)\Big)_{(i_1,i_2,k)}$ as a bipartite graph G. One partite set of G — shown in the upper half — corresponds to F_1 and contains a vertex for each component of F_1. Similarly for the other partite set shown in the lower half corresponding to F_2. In the upper partite set, there are s_{1,i_1} vertices corresponding to copies of S_{i_1}. The edges of G encode which root vertices of components of F_1 can be mapped onto which root vertices of components of F_2. Since we aim for a nice common subgraph, the edges of G reflect $c_2 = 2$.

Therefore, we may restrict ourselves, for each $i_1 \in [q]$, to $O\left(\frac{(2c_2+1)o_{\max}\Delta}{\epsilon}\right)$ different values for each $s(i_1, i_2, k)$, which, for fixed ϵ, yields constantly many choices, say at most c_3, for

$$M(i_1) = \Big(s(i_1, i_2, k)\Big)_{(i_2,k)\in[q]\times\left[o_{(i_1,i_2)}\right]:|i_2-i_1|\leq c_2}.$$

Since $q \leq c_1 \log(n)$, this results in at most

$$c_3^{c_1 \log(n)} = n^{c_1 \log(c_3)}$$

many choices for

$$M = \Big(s(i_1, i_2, k)\Big)_{(i_1,i_2,k)\in[q]^2\times\left[o_{(i_1,i_2)}\right]:|i_2-i_1|\leq c_2},$$

that is, polynomially many. Note that such an M is compatible with the instance F_1 and F_2 if

$$\begin{aligned} s_{1,i_1} - \sum_{i_2\in[q]:|i_2-i_1|\leq c_2} \; \sum_{k\in\left[o_{(i_1,i_2)}\right]} s(i_1, i_2, k) &\geq 0 \text{ for every } i_1 \in [q], \text{ and} \\ s_{2,i_2} - \sum_{i_1\in[q]:|i_2-i_1|\leq c_2} \; \sum_{k\in\left[o_{(i_1,i_2)}\right]} s(i_1, i_2, k) &\geq 0 \text{ for every } i_2 \in [q]. \end{aligned}$$

In fact, these two differences count the number of roots of components of F_1 and F_2, respectively, that are either of degree less than $\frac{\Delta}{\epsilon}$ or are of larger degree and correspond to an isolated vertex in the nice solution F.

See Figure 4 for an illustration.

Putting things together

For each of the polynomially many compatible choices for M,

- removing, for every $i_1, i_2 \in [q]$ with $|i_2 - i_1| \leq c_2$ and $k \in \left[o_{(i_1,i_2)}\right]$, from $s(i_1, i_2, k)$ pairs (K_1, K_2) of components $K_1 \cong S_{i_1}$ of F_1 and $K_2 \cong S_{i_2}$ of F_2 the parts $A_{0,1}$ and $A_{0,2}$ isomorphic to the corresponding subtrees A_0 as encoded by $y^k_{(i_1,i_2)} \in \tilde{Y}_{(i_1,i_2)}$, and
- removing all edges incident with roots of degree at least $\frac{\Delta}{\epsilon}$ within the remaining components of F_1 and F_2,

results in subforests F_1^M of F_1 and F_2^M of F_2 whose components all have orders less than $1 + \frac{\Delta^2}{\epsilon}$. Let F' denote the union of all parts isomorphic to the corresponding subtrees A_0 removed from F_1. Using Lemma 2, we can determine in polynomial time a common subgraph F'' of F_1^M and F_2^M with $\mathrm{lcs}(F_1^M, F_2^M)$ edges and $F'' \subseteq F_1^M$. Now, $F' \cup F''$ is a common subgraph of F_1 and F_2 that is compatible with M and has the maximum possible number of edges subject to this condition. As explained along the proof, considering only the polynomially many choices for M will produce a common subgraph F^* of the form $F' \cup F''$ such that $m(F^*) \geq \mathrm{lcs}(F_1, F_2) - C\epsilon n$ for some fixed integer C independent of ϵ, which completes the proof. □

References

[1] T. Akutsu. An RNC Algorithm for Finding a Largest Common Subtree of Two Trees. *IEICE Transactions on Information and Systems*, E75-D:95–101, 1992.

[2] T. Akutsu and M. Halldórsson. On the approximation of largest common subtrees and largest common point sets. *Theoretical Computer Science*, 233:33–50, 2000.

[3] T. Akutsu and T. Tamura. On the Complexity of the Maximum Common Subgraph Problem for Partial k-Trees of Bounded Degree. *Lecture Notes in Computer Science*, 7676:146–155, 2012.

[4] S.H. Bokhari. On the Mapping Problem. *IEEE Transactions on Computers*, C-30:207–214, 1981.

[5] A. Droschinsky, N.M. Kriege, and P. Mutzel. Faster algorithms for the maximum common subtree isomorphism problem. *Leibniz International Proceedings in Informatics*, 58:Art. No. 33, 14 pp., 2016.

[6] A. Droschinsky, N.M. Kriege, and P. Mutzel. Largest weight common subtree embeddings with distance penalties. *Leibniz International Proceedings in Informatics*, 117:Art. No. 54, 15 pp., 2018.

[7] M.R. Garey and D.S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco, 1979.

[8] M. Grohe, G. Rattan, and G.J. Woeginger. Graph Similarity and Approximate Isomorphism. *Leibniz International Proceedings in Informatics*, 117:Art. No. 20, 16 pp., 2018.

[9] E. de Gastines and A. Knippel. Formulations for the maximum common edge subgraph problem. *Discrete Applied Mathematics*, 346:115–130, 2024.

[10] V. Kann. On the approximability of the maximum common subgraph problem. *Lecture Notes in Computer Science*, 577:377–388, 1992.

[11] S. Khanna, R. Motwani, and F.F. Yao. Approximation Algorithms for the Largest Common Subtree Problem. *Technical report*, Stanford University, CS-TR-95-1545.

[12] N. Kriege, F. Kurpicz, and P. Mutzel. On maximum common subgraph problems in series-parallel graphs. *European Journal of Combinatorics*, 68:79–95, 2018.

[13] D.W. Matula. Subtree Isomorphism in $O\left(n^{5/2}\right)$. *Annals of Discrete Mathematics*, 2:91–106, 1978.

[14] J.W. Raymond and P. Willett. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *Journal of Computer-Aided Molecular Design*, 16:521–533, 2002.

[15] K. Shearer, H. Bunke, and S. Venkatesh. Video indexing and similarity retrieval by largest common subgraph detection using decision trees. *Pattern Recognition*, 34:1075–1091, 2001.

[16] R.P. Stanley. Catalan numbers. Cambridge University Press, New York, 2015.

[17] A. Yamaguchi, K.F Aoki, and H. Mamitsuka. Finding the maximum common subgraph of a partial k-tree and a graph with a polynomially bounded number of spanning trees. *Information Processing Letters*, 92:57–63, 2004.